融合差分变异和切线飞行的天鹰优化器 *

徐亦凤，刘　升[†]，刘宇凇，张伟康

(上海工程技术大学 管理学院，上海 201620)

摘　要：针对天鹰优化器(aquila optimizer，AO)虽然拥有强大的全局勘探能力，但局部开发能力不足的问题，提出融合差分变异和切线飞行的天鹰优化器(differential evolution mutation and tangent flight aquila optimizer，DEtanAO)。首先，根据差分进化算法中的变异操作能使算法具有较强的开发能力弥补了 AO 算法的不足，然后，利用切线搜索算法中切线飞行策略具有较强的探索搜索空间的能力并能使算法跳出局部最优解的优势，用其替换了 AO 算法中的莱维飞行。这两种策略的结合有效地平衡了 DEtanAO 算法的勘探和开发阶段。最后，为验证 DEtanAO 算法的优化性能，在 12 个标准基准函数、高维函数、Wilcoxon 秩和检验和工程优化问题上来测试所改进算法的寻优能力。实验结果表明，相比其他新提出的智能算法，DEtanAO 算法具有更强的寻优能力和更快的收敛速度。

关键词：天鹰优化器；差分变异；切线飞行；勘探与开发

中图分类号：TP301.6　　doi: 10.19734/j.issn.1001-3695.2022.03.0129

Aquila optimizer integrating differential mutation and tangent flight

Xu Yifeng, Liu Sheng[†], Liu Yusong, Zhang Weikang

(*School of Management*, *Shanghai University of Engineering Science*, *Shanghai* 201620, *China*)

Abstract: The Aquila optimizer (AO) , although capable of robust global exploration, had problems with inadequate local development. In the study, a differential evolution mutation and tangent flight Aquila optimizer (DEtanAO) was proposed. Firstly, the mutation operation in the differential evolution algorithm can make the algorithm have strong development ability and make up for the shortcomings of the AO algorithm. Then, the tangent flight strategy in the tangent search algorithm has a strong ability to explore the search space and can make the algorithm jump out of the local optimal solution, which is used to replace the Levy flight in the AO algorithm. The combination of these two strategies effectively balanced the exploration and exploitation stage of the DEtanAO algorithm. Finally, in order to verify the optimization performance of the DEtanAO algorithm, the optimization ability of the improved algorithm was tested in 12 standard benchmark functions, high-dimensional functions, Wilcoxon rank-sum test, and engineering optimization problems. The results showed that the optimization ability and convergence speed of the DEtanAO algorithm was better, compared with other newly proposed intelligent algorithms.

Key words: aquila optimizer; differential evolution mutation; tangent flight; exploration and exploitation

0　引言

随着现实世界中离散、无约束性质的优化问题越来越复杂，传统的基于数学的优化算法很难找到最优解，并且存在一定的缺点和局限性，例如收敛至局部最优和未知的搜索空间并且对于优化问题的解决方案单一等。为解决上述问题，近年来，基于仿生学的元启发式优化算法不断被提出和改进，并以其操作简便、参数调整简单等特点大受学者青睐，如：加入自适应权重模拟鸟群觅食行为的混沌粒子群算法[1](Chaotic-Particle Swarm Optimization，CPSO)，受麻雀觅食和反哺行为启发提出的麻雀搜索算法[2](Sparrow Search Algorithm, SSA)，受海洋适者生存理论启发提出的海洋捕食者算法[3](Marine Predators Algorithm,MPA)，融合了精英反向和黄金正弦策略的鲸鱼优化算法[4](Elite Opposition Based Golden-Sine Whale Optimization Algorithm，EGolden-SWOA)等等。这些算法的提出为解决复杂的优化问题提供了更多的思路。

天鹰优化器(Aquila Optimizer，AO)[5]是 Laith Abualigah 等人于 2021 年提出了一种新的基于种群的优化方法，该方法受到自然界中天鹰在捕获猎物过程中行为的启发，具有强大的全局勘探能力、搜索效率高、收敛速度快等优点，但其局部开发能力不足，容易陷入局部最优。王爽等[6]提出了一种改进的混合天鹰优化器和哈里斯鹰眼优化器(IHAOHHO)，将 HHO 算法的开发策略集成到 AO 算法的探索策略中，并添加反向学习策略(ROBL)以避免局部最优停滞。同时，非线性逃逸能量参数平衡了算法的探索和开发阶段。这些改进使得 IHAOHHO 算法的寻优性能得到提升；除此之外，Shubham Mahajan[7]等人将 AO 算法和算数优化算法相结合(AO-AOA)来解决优化问题，并在 23 个基准函数上验证了所提算法的有效性与优越性；王世成[8]等人提出了一种改进的多目标天鹰优化器，并用其设计了一款基于混合固体氧化物燃料电池(SOFC)系统的混合动力系统，从而改善了整体成本和用火效率；Eric Ofori-Ntow Jnr[9]等人将离散小波变换(DWT)、混沌理论的相空间重构(PSR)、天鹰优化算法(AOA)和反向传播神经网络(BPNN)混合起来提出了 DWT-PSR-AOA-BPNN 模型，并将其应用于预测风速，实验结果表明该模型预测精度较高。但是由于天鹰优化器提出的时间较短，国内并没有对 AO 算法的改进进行充分研究，其性能仍有进一步的提升空间以便应用于更多的实际问题中。本文提出一

收稿日期：2022-03-16；**修回日期：**2022-05-16　　**基金项目：**国家自然科学基金资助项目(61673258，61075115)；上海市自然科学基金资助项目(19ZR1421600)

作者简介：徐亦凤(1998-)，女，江苏无锡人，硕士研究生，主要研究方向为智能算法、金融(859148037@qq.com)；刘升(1966-)，男(通信作者)，湖北黄石人，教授，硕导，主要研究方向为智能计算、群智能系统、进化算法(ls6601@sina.com)；刘宇凇(1997-)，男，内蒙古包头人，硕士研究生，主要研究方向为智能算法；张伟康(1996-)，男，山东临沂人，硕士研究生，主要研究方向为商务统计、智能计算.

种融合差分变异和切线飞行的天鹰优化器(DEtanAO)，首先在 AO 算法中加入差分变异策略，提高算法的搜索精度，接着利用切线搜索算法[10](Tangent Search Algorithm，TSA)中的切线飞行策略替代 AO 算法中的莱维飞行策略，进一步优化 AO 的寻优性能，使其跳出局部最优解。为证明改进算法的有效性，随后在单模态、多模态和高维测试函数上进行多种算法的对比实验，并进行 Wilcoxon 秩和检验分析，进一步在工程应用问题上检验所提算法的前沿性，结果表明 DEtanAO 算法的寻优精度、收敛速度有着显著提升。

1　天鹰优化器

天鹰优化器模拟了天鹰对不同猎物的不同捕猎方式。天鹰对于快速移动猎物的狩猎方式反映了算法的全局探索能力，对于慢速移动猎物的狩猎方式反映了算法的局部开发能力。AO 算法具有较强的全局探索能力、较高的搜索效率和较快的收敛速度，但其局部开发能力不足，容易陷入局部最优。AO 算法模拟了天鹰在狩猎期间的行为，其优化过程用四种方法表示：通过垂直弯腰的高翱翔选择搜索空间 (X_1)；通过短滑翔攻击的等高飞行在搜索空间内探索 (X_2)；通过慢速下降攻击的低空飞行在收敛搜索空间内探索 (X_3)，以及通过行走和抓取猎物进行俯冲 (X_4)。

1.1　拓展探索 (X_1)

在第一种方法 (X_1) 中，天鹰识别猎物区域并通过垂直弯腰的高翱翔来选择最佳狩猎区域。在这里，天鹰从高空翱翔来确定搜索空间的区域即搜索猎物在哪里。图 1 显示了天鹰高翱翔和垂直弯腰的行为。

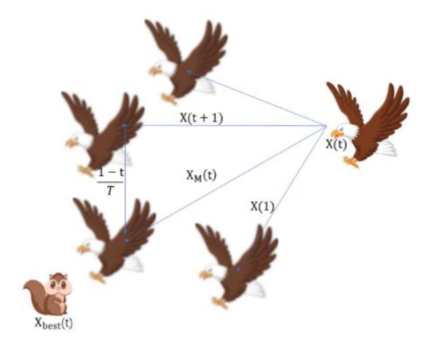


图 1　天鹰高翱翔和垂直弯腰的行为

Fig. 1　The behavior of the Aquila high soar with the vertical stoop

这种行为用数学式(1)表示为

$$X_1(t+1)=X_{best}(t)\times\left(1-\frac{t}{T}\right)+(X_M(t)-X_{best}(t))\times rand \tag{1}$$

$$X_M(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t),\forall j=1,2,\cdots,Dim \tag{2}$$

其中，$X_1(t+1)$ 是由第一个搜索方法 (X_1) 生成的 t 的下一次迭代的解，$X_{best}(t)$ 是第 t 次迭代的最优解，这反映了猎物的大致位置。$(1-t/T)$ 表示通过迭代次数控制探索，$X_M(t)$ 表示当前解决方案在第 t 次迭代时的平均值。rand 是[0,1]之间的随机数。t 和 T 分别表示当前迭代次数和最大迭代次数。Dim 是问题的维度，N 是候选解的个数。

1.2　缩小探索范围 (X_2)

在第二种方法 (X_2) 中，当天鹰从高空找到猎物区域时，会在目标猎物上方盘旋，准备好发动攻击。这种方法称为短滑翔攻击的等高飞行。此时，天鹰优化器狭窄地探索目标猎物的选定区域，为攻击做准备。图 2 展示了带有短滑翔攻击的天鹰等高飞行的行为。这种行为用数学式(3)表示为

$$X_2(t+1)=X_{best}(t)\times levy(D)+X_R(t)+(y-x)\times rand \tag{3}$$

$$levy(D)=s\times\frac{u\times\sigma}{|v|^{\frac{1}{\beta}}} \tag{4}$$

$$\sigma=\left(\frac{\Gamma(1+\beta)\times\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\times\beta\times2^{\left(\frac{\beta-1}{2}\right)}}\right) \tag{5}$$

其中，$X_2(t+1)$ 是由第二个搜索方法 (X_2) 生成的 t 的下一次迭代的解，D 表示维度空间，$levy(D)$ 是莱维飞行分布函数，$X_R(t)$ 是取值范围在[1,N]的随机解。其中，s 为固定为 1.5 的常量值，y 和 x 在搜索中呈现螺旋形式，其计算公式如下：

$$y=r\times\cos(\theta) \tag{6}$$

$$x=r\times\sin(\theta) \tag{7}$$

$$r=r_1+U\times D_1 \tag{8}$$

$$\theta=-\omega\times D_1+\theta_1 \tag{9}$$

其中，$\theta_1=(3\times\pi)/2$，r_1 为介于 1 和 20 之间的固定周期指数，U 的值为 0.00565，D_1 是 1 到搜索空间长度的整数，ω 的值为 0.005。

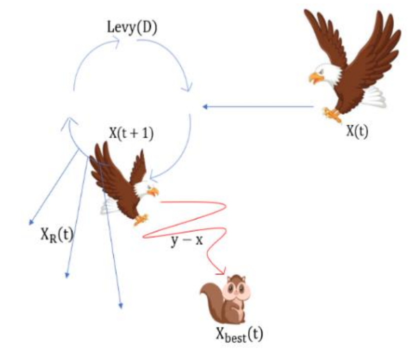


图 2　天鹰短滑翔攻击的等高飞行的行为

Fig. 2　The behavior of the Aquila contour flight with short glide attack

1.3　扩大开发 (X_3)

在第三种方法 (X_3) 中，当天鹰锁定了捕食区域，天鹰准备好着陆和攻击，随后垂直下降并进行初步攻击来试探猎物反映，这种行为称为低空飞行和慢速下降攻击。图 3 展示了天鹰低空飞行和缓慢下降的攻击行为。

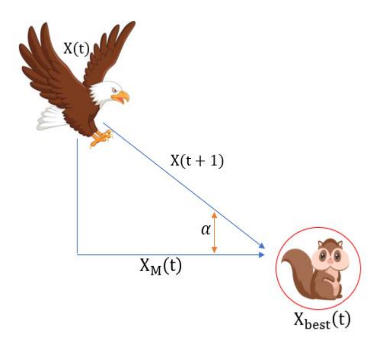


图 3　天鹰低空飞行和缓慢下降的攻击行为

Fig. 3　The behavior of the Aquila low flight with slow descent attack

这种行为用数学式(10)表示为

$$X_3(t+1)=(X_{best}(t)-X_M(t))\times\alpha-rand+((UB-LB)\times rand+LB)\times\delta \tag{10}$$

其中，$X_3(t+1)$ 是由第三个搜索方法 (X_3) 生成的 t 的下一次迭代的解，α 和 δ 是开采调整参数，其值较小，在(0,1)的范围内，LB、UB 分别代表所给问题的上下限。

1.4　缩小开发范围 (X_4)

在第四种方法 (X_4) 中，当天鹰接近猎物时，它会根据猎物的随机移动攻击猎物，这种方法称为“行走并抓住猎物”如图 4 所示。

这种行为用数学式(11)表示为

$$X_4(t+1)=QF\times X_{best}(t)-(G_1\times X(t)\times rand)-G_2\times levy(D) \tag{11}$$

$$QF(t)=t^{\frac{2\times rand-1}{(1-T)^2}} \tag{12}$$

$$G_1=2\times \text{rand-1} \tag{13}$$

$$G_2=2\times\left(1-\frac{t}{T}\right) \tag{14}$$

其中，$X_4(t+1)$ 是由第四个搜索方法 (X_4) 生成的 t 的下一次迭代的解，QF 表示用于平衡搜索策略的质量函数，其中 $F\in(0,1)$，G_1 表示猎物在逃逸过程中天鹰采用的不同方法，G_2 表示从 2 到 0 的递减值，代表了天鹰在追踪猎物时从第一个位置到最后一个位置时的飞行坡度。$QF(t)$ 是第 t 次迭代时的质量函数值，rand 是(0,1)的随机数，t 和 T 分别表示当前和最大的迭代次数。图 5 是 AO 算法流程。

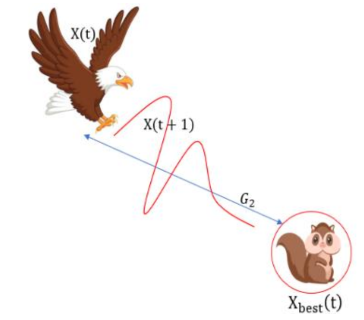


图 4　行走并抓住猎物进行俯冲的行为

Fig. 4　Behavior of the Aquila walk and grab prey

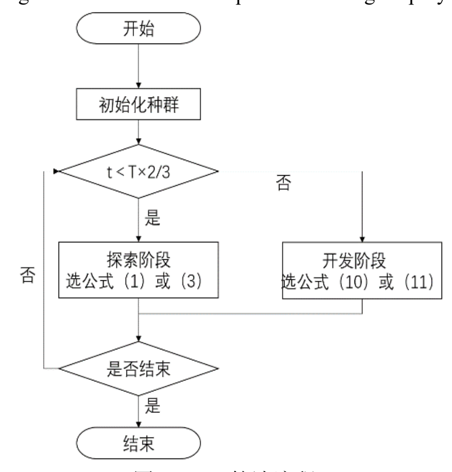


图 5　AO 算法流程

Fig. 5　The flow chart of AO algorithm

在 AO 算法的两个阶段，都需要添加贪心算法来保证个体的新位置优于其原位置。从 AO 算法的实现公式来看，四个位置更新公式中，式(1)和(11)会向 0 收敛，式(10)会收敛到常数，只有式(3)勉强会收敛至最优解，故 AO 算法易陷入局部最优。

2　改进的天鹰优化器(DEtanAO)

2.1　差分变异策略

差分进化算法[11](DE)是一种基于种群进化的算法。通过对种群采取三种进化操作：变异操作、交叉操作以及选择操作进行迭代使算法趋于全局最优解。DE 算法利用差分这种变异操作来产生变异种群作为新的个体；再通过交叉操作，对变异种群和原始种群进行交叉，得到交叉种群；对原始种群和交叉种群利用贪婪选择选取下一代种群。差分变异指的是 DE 通过两个不同父代向量做差，实现变异个体与差分向量的结合，常用的变异策略如表 1 所示。

表 1 中 $F\in(0,1)$ 为缩放因子，$r_1\neq r_2\neq r_3$ 且 r_1、r_2、r_3 为随机整数，$X^t_{best,j}$ 表示第 j 代最优个体，$V^t_{i,j}$ 表示变异后的个体，$X^t_{i,j}$ 表示第 j 代第 i 个个体，$X^t_{r_1,j}$、$X^t_{r_2,j}$、$X^t_{r_3,j}$ 表示当前种群中随机选取的不同个体。

基于“best”的变异策略具有良好的开发能力和快速收敛的性能，以天鹰当前种群中适应度最好的个体作为基矢量，式中 $X^t_{r_2,j}$、$X^t_{r_3,j}$ 由天鹰种群随机产生。其中，变异算子 F 随着迭代次数的增加动态随机变化，在算法迭代初期，F 值较小时，缩放因子对变异的影响较小，天鹰的变异受群体中最优天鹰的影响，变异的天鹰会快速朝向群体最优个体靠近；在算法运行后期，F 值较大时，缩放因子对变异影响较大，天鹰聚集在最优天鹰个体的周围，使得种群在最优个体周围进行局部细致探索，提高算法的搜索精度。

表 1　变异策略

Tab. 1　Variation strategy

变异策略	数学表达式
DE/rand/1	$V^t_{i,j}=X^t_{r_1,j}+F\left(X^t_{r_2,j}-X^t_{r_3,j}\right)$
DE/best/1	$V^t_{i,j}=X^t_{best,j}+F\left(X^t_{r_2,j}-X^t_{r_3,j}\right)$
DE/current-to-best/1	$V^t_{i,j}=X^t_{r_1,j}+F\left(X^t_{best,j}-X^t_{i,j}\right)+F\left(X^t_{r_1,j}-X^t_{r_2,j}\right)$

2.2　切线飞行策略

切线搜索算法[10](The Tangent Search Algorithm，TSA)是 2021 年提出的一种新的基于种群的优化算法。TSA 使用一个基于切线函数的数学模型将给定的解移向更好的解。无论是基于导数还是基于微分的最优化算法都是类似于下式(15)的下降等式：

$$X^{t+1}=X^t+step*d \tag{15}$$

其中 step 是移动的大小，d 是移动的方向。最优化算法的区别在于如何计算步长，基于导数的方法使用梯度下降法或海森矩阵；然而基于微分的方法中，例如元启发式算法，它们使用随机步骤收敛到全局最优。比如，遗传算法[12]使用高斯变异作为步长；差分进化算法[13]通过当前种群个体之间的差值来计算步长；布谷鸟搜索算法[14]是通过莱维飞行函数来计算步长。

步长的优化对于算法的优化极为关键，步长的大值有利于探索，小值有利于开发。TSA 算法中提出了一种基于切线函数的新的步长，称之为切线飞行，切线函数有助于有效地探索搜索空间。全局和局部游走相结合的探索方程由下式(16)所表示。

$$X^{t+1}=X^t+step*\tan(\theta) \tag{16}$$

从数学逻辑上讲，越接近 $\pi/2$ 的 θ，切值越大，得到的解离当前解越远；越接近 0 的 θ，切值越小，得到的解越接近当前解。探索搜索方程应用于概率为 1/D 的每个变量，其中 D 为问题的维数。

用切线飞行函数替代天鹰优化器中的莱维飞行函数使 AO 在规避局部开发不足的同时又进一步提高 AO 算法的全局探索能力，图 6、7 是用 Mantegna 方法[15]模拟 1000 次的莱维飞行和切线飞行的随机游走示意图。

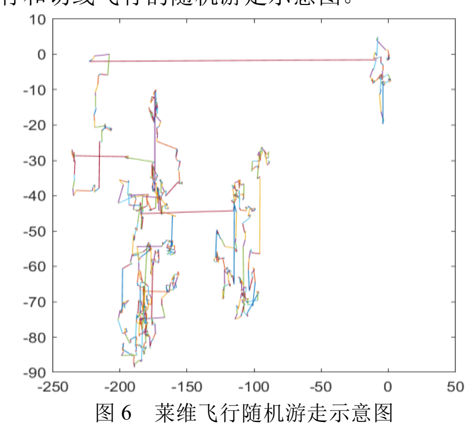


图 6　莱维飞行随机游走示意图

Fig. 6　A diagram of the Levy Flight random walk

从图 6、7 可以看出莱维飞行产生的步幅随机性大、步长区间窄，导致传统算法可能出现在迭代前期搜寻距离过小，而在迭代末期搜寻出现搜寻距离过大的问题，使得算法优化迭代周期过长，精度不足；然而切线飞行出现大跨步的频率要比莱维飞行高，弥补了莱维飞行搜寻距离过大过小的问题，

切线飞行比莱维飞行更有利于跳出局部最优解并进行大范围的搜索，使得天鹰获得更多的捕食机会。

2.3 DEtanAO 的时间复杂度分析

时间复杂度分析通常依赖于三个规则：种群初始化、计算适应度函数和更新解。假设种群数量为N，O(N)是种群初始化过程的计算复杂性，O(T×N)+O(T×N×Dim)是解的更新过程的计算复杂性，其中，总迭代次数称为 T，问题维度为 Dim。因此，AO 的总的时间复杂度为 O(N×(T×D+1))。

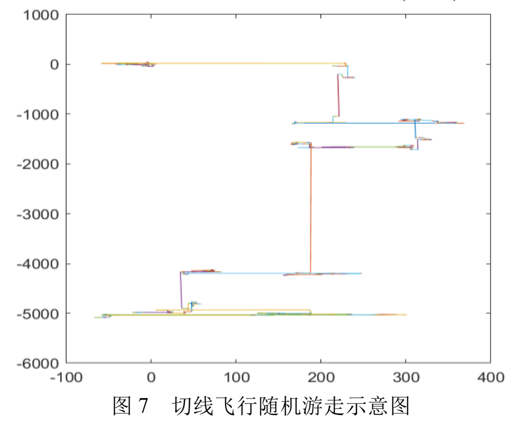


图 7 切线飞行随机游走示意图

Fig. 7 Tangential flight random walk schematic

由标准 AO 的复杂度可知，DEtanAO 仅添加了差分进化算法中的变异策略 O(T×N)，并将原来的 Levy 飞行策略变成 tangent 飞行策略，不增加计算复杂性。因此，DEtanAO 的总的时间复杂度为 O(N×(T×(D+1)+1))。

综上分析，DEtanAO 算法复杂度与标准 AO 算法是时间复杂度属于同一级别，并未增加额外的计算阶级。DEtanAO 算法的伪代码如下所示。

算法 Pseudo-code of the DetanAO

```
输入: i,e,α,δ,etc
输出: best position、best fitness
Initialization the population %种群初始化
Define initial parameter( i,e,α,δ,etc ) %定义初始参数
while(The end condition is not met) do
  Calculate the fitness function values and the best obtained solution X_best(t) . %计算最佳适应度值
  for (i=1,2,…, N) do
    Update the X_M(t),x,y,G_1,G_2,Levy(D),etc %更新各参数
    Expanded exploration (X_1) : Update the current solution using Equation(1) %拓展探索：用式(1)更新位置1
    Narrowed exploration (X_2) : Update the current solution using Equation(3) %缩小探索范围：用式(3)更新位置2
    Expanded exploitation (X_3) : Update the current solution using Equation(10) %扩大开发：用式(10)更新位置3
    Narrowed exploitation (X_4)：Update the current solution using Equation(11) %缩小开发范围：用式(11)更新位置4
  end for
  for
    Mutation:generate a mutant population %差分变异策略
  end for
end while
function tangent flight %切线飞行策略
end
```

3 仿真实验与结果分析

3.1 仿真实验环境

本次仿真测试环境为：操作系统版本为Win10、64位操作系统，处理器为 Intel® Core™ i5-10210U CPU @ 1.60GHz 2.11 GHz，内存 16.0GB，主频 2.11GHz，仿真软件为 MATLAB 2020b。

3.2 比较对象和参数设置

本文选取基本天鹰优化器(AO)[5]、混沌粒子群算法(CPSO)[1]、麻雀搜索算法(SSA)[2]、海洋捕食者算法(MPA)[3]、融合了精英反向和黄金正弦策略的鲸鱼优化算法(EGolden-SWOA)[4]以及用切线飞行策略替换莱维飞行的天鹰优化器(tanAO)，与本文所提的融合差分进化和切线飞行的天鹰优化器(DEtanAO)进行对比。为保证实验的公平有效，所有算法的种群规模设置为 30，迭代次数设置为 500，其他参数设置如表 2 所示。

表 2 各算法的实验参数

Tab. 2 Experimental parameters of each algorithm

算法	参数
AO	α =0.1，δ =0.1，U =0.0056，ω =0.005
tanAO	α =0.1，δ =0.1，U =0.0056，ω =0.005
DEtanAO	α =0.1，δ =0.1，U =0.0056，ω =0.005
CPSO	C_1=1.5，C_2=1.5，u=2
SSA	ST=0.8，PD=0.2，SD=0.2
MPA	P=0.5，FADs=0.2
EGolden-SWOA	b=1

3.3 测试函数

为验证改进算法有更好的寻优性能，本文选取 12 个基准测试函数进行函数优化对比实验，其中选取单峰测试基准函数(f_1 ~ f_6)用于检查算法的局部开发能力，多峰测试基准函数(f_7 ~ f_9)用于检查算法的全局勘探能力，固定维度多峰函数(f_{10} ~ f_{12})，具体函数信息见表 3。

表 3 基准测试函数

Tab. 3 Benchmark function

编号	函数	维数	范围	最优值
	Sphere	dim	[-100,100]	0
f_2	Schwefel 2.22	dim	[-30,30]	0
f_3	Schwefel 1.2	dim	[-100,100]	0
f_4	Schwefel 2.21	dim	[-100,100] f_1	0
f_5	Rosenbrock	dim	[-30,30]	0
f_6	Quartic	dim	[-1.28,1.28]	0
f_7	Rastrigin	dim	[-5.12,5.12]	0
f_8	Ackley	dim	[-32,32]	0
f_9	Penalized	dim	[-50,50]	0
f_{10}	Foxholes	2	[-65,65]	1
f_{11}	Branin	2	[-5,5]	0.398
f_{12}	Goldstein Price	2	[-2,2]	3

3.4 寻优精度分析

将本文所提出的 DEtanAO，与 AO、tanAO、CPSO、SSA、MPA、EGolden-SWOA 分别在 12 个基准函数上独立运行 30 次，其中非固定维度函数的维度设置为 dim=30/100/500。表 4、5 统计了 7 种算法分别在低维、高维和固定维数时，得到的最优解、平均值和标准差。

表 4 的测试结果显示，DEtanAO 在函数 f_1、f_3、f_{10}、f_{12} 上的寻优效果达到 100%，可以直接搜索到对应函数的最优值。相较于其他对比算法，DEtanAO 在函数 f_1、f_2、f_3、f_4、f_8 上的标准差为0，说明 DEtanAO 具有较强的鲁棒性，差分变异策略保证了种群的多样性。在低维条件下，除了直接能达到最优值的函数，DEtanAO 相比其他对比算法至少高出 11 个数量级、至多高出 235 个数量级。在高维条件下，DEtanAO 相比其他对比算法在求解大规模函数时，仍能保持较高水平，最大程度的接近函数最优值，其中 f_3 和 f_7 仍能直接找到最优解，f_8 保持与低维条件下相同的水平，表

明所提算法在求解大规模问题时未陷入“维数灾难”，具有较强的稳定性。根据表 5，在固定维数下，MPA 虽然表现最好，但是 DEtanAO 仍可以直接搜寻到最优解，从标准差来看，鲁棒性也较好。综上所述，DEtanAO 的稳定性以及精度均较其他算法有大幅度提升。这说明，DEtanAO 通过引入差分变异策略改善了原算法过度依赖天鹰根据不同攻击方式的数学公式进行位置更新的状况，利用切线飞行公式，使得算法跳出局部最优，提高了算法的寻优质量。

表 4　低维、高维函数上的性能比较

Tab. 4　Performance comparison on low-dimensional and high-dimensional functions

函数	算法	dim=30			dim=100			dim=500		
		最优解	平均值	标准差	最优解	平均值	标准差	最优解	平均值	标准差
f_1	DEtanAO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	tanAO	6.34E-174	1.18E-107	4.26E-108	9.01E-167	1.67E-117	9.12E-117	3.47E-163	4.70E-108	2.25E-107
	AO	1.32E-160	3.45E-105	1.89E-106	1.61E-157	1.15E-98	6.29E-98	2.33E-160	3.88E-103	2.13E-102
	CPSO	4.98E-06	1.0769	2.7904	2.24E-06	0.68775	2.1899	1.22E-06	0.54379	2.0649
	SSA	0	3.15E-58	1.63E-57	3.43E-143	4.45E-56	2.42E-55	0	5.28E-56	2.85E-55
	MPA	2.54E-24	5.38E-23	7.51E-23	8.95E-21	2.54E-19	2.25E-19	1.26E-17	8.00E-17	4.93E-17
	EGolden-SWOA	0	0	0	0	0	0	0	0	0
f_2	DEtanAO	**8.86E-234**	**1.50E-210**	**0**	**6.15E-231**	**2.19E-201**	**0**	**1.23E-228**	**4.87E-206**	**0**
	tanAO	1.46E-81	2.61E-54	1.42E-53	1.04E-80	3.46E-52	1.33E-51	2.60E-78	1.96E-53	1.07E-52
	AO	3.59E-80	2.09E-74	4.49E-74	1.02E-79	3.24E-51	1.77E-50	1.80E-78	2.43E-55	1.33E-54
	CPSO	2.88E-07	0.28538	1.5558	4.08E-05	0.26831	1.4591	2.52E-05	0.26964	1.0202
	SSA	6.66E-177	7.29E-28	3.99E-27	2.66E-88	8.45E-26	4.30E-25	6.99E-78	7.82E-31	3.03E-30
	MPA	1.44E-14	2.45E-13	1.95E-13	1.74E-12	1.87E-11	1.87E-11	4.51E-11	6.19E-10	9.86E-10
	EGolden-SWOA	1.38E-220	8.98E-174	0	1.13E-208	3.65E-171	0	2.52E-204	4.88E-161	2.67E-160
f_3	DEtanAO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	tanAO	1.76E-159	1.17E-101	6.41E-101	3.27E-160	1.29E-101	7.09E-98	5.14E-101	5.01E-104	2.74E-103
	AO	4.96E-154	7.05E-103	3.67E-102	2.98E-155	2.08E-98	1.09E-100	5.20E-98	6.21E-98	3.37E-97
	CPSO	2.41E-09	0.52855	1.7034	3.61E-05	0.90048	2.4076	6.03E-07	0.26886	1.018
	SSA	1.65E-299	6.72E-29	2.94E-28	0	4.15E-23	2.26E-22	0	5.22E-23	2.25E-22
	MPA	2.54E-08	0.0001959	0.0005031	0.0018476	13.018711	36.046243	165.03471	5865.8164	4933.6315
	EGolden-SWOA	0	0	0	0	7.47E-286	0	0	2.43E-273	0
f_4	DEtanAO	**2.75E-235**	**1.47E-216**	**0**	1.45E-237	5.05E-217	0	**1.25E-234**	**1.00E-220**	**0**
	tanAO	2.89E-81	5.25E-54	2.87E-53	4.08E-81	4.48E-55	2.45E-54	1.69E-80	2.81E-53	1.54E-52
	AO	5.25E-82	6.87E-55	3.76E-54	1.50E-81	9.32E-57	5.10E-56	4.57E-80	6.24E-62	3.42E-61
	CPSO	1.17E-07	1.2247	2.8645	2.82E-06	0.51696	1.6558	5.44E-07	0.95575	2.5679
	SSA	0	9.95E-34	5.15E-33	5.99E-160	2.04E-32	1.11E-31	1.73E-74	1.08E-31	4.32E-31
	MPA	5.24E-10	3.23E-09	1.47E-09	4.65E-08	2.29E-07	1.36E-07	5.83E-06	2.23E-05	1.32E-05
	EGolden-SWOA	4.15E-211	2.21E-171	0	**0**	**7.47E-286**	**0**	1.17E-206	2.92E-179	0
f_5	DEtanAO	**4.34E-28**	0.0001801	0.0006804	**6.40E-28**	2.16E-05	4.89E-05	**1.75E-26**	**1.01E-06**	**2.75E-06**
	tanAO	1.01E-06	0.0022965	0.0038784	3.81E-06	0.0016667	0.0053777	1.00E-05	0.0051573	0.0066908
	AO	1.35E-06	0.0060161	0.0108928	0.0001869	0.0232199	0.0363135	1.05E-06	0.1357954	0.2335892
	CPSO	4.20E-06	0.54473	2.0528	7.86E-06	0.27242	1.4861	1.11E-05	0.54892	1.7798
	SSA	1.33E-08	3.46E-05	9.84E-05	1.49E-06	0.0071157	0.0228404	6.24E-07	0.0179906	0.0352481
	MPA	24.233312	25.357126	0.5291385	95.903627	97.181244	0.6738597	496.48333	497.07268	0.1888052
	EGolden-SWOA	2.27E-11	**6.69E-06**	**3.55E-05**	2.63E-09	**8.95E-07**	**1.89E-06**	2.39E-09	8.36E-05	0.0004372
f_6	DEtanAO	7.78E-06	0.0001963	0.0001794	8.70E-06	0.0001509	0.0001348	1.32E-05	0.0001044	6.30E-05
	tanAO	2.69E-06	7.40E-05	5.62E-05	1.10E-05	0.0001021	9.00E-05	1.12E-05	9.67E-05	7.52E-05
	AO	4.87E-06	8.34E-05	7.30E-05	**4.76E-07**	**8.42E-05**	**7.89E-05**	**7.42E-06**	**7.20E-05**	**5.09E-05**
	CPSO	9.69E-09	0.82239	2.2742	4.61E-06	0.27941	1.0169	2.58E-05	0.56843	2.1232
	SSA	9.10E-05	0.0017566	0.0017836	0.0001687	0.0017752	0.0012219	0.0001651	0.0033734	0.0050135
	MPA	0.0002901	0.0013413	0.0006722	0.0003923	0.0018288	0.0008126	0.0006264	0.0022338	0.0009748
	EGolden-SWOA	**3.32E-06**	**8.59E-05**	**7.24E-05**	5.67E-07	0.0001001	0.0001069	8.48E-06	9.47E-05	7.95E-05
f_7	DEtanAO	**0**	3.79E-15	2.08E-14	**0**	**0**	**0**	**0**	**0**	**0**
	tanAO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	AO	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	CPSO	2.05E-06	0.54097	2.0541	1.05E-08	0.2739	1.4954	8.63E-08	0.54843	2.0851
	SSA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	MPA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	EGolden-SWOA	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
f_8	DEtanAO	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**
	tanAO	8.88E-16	8.88E-16	0	8.88E-16	2.52E-13	1.37E-12	8.88E-16	8.88E-16	0
	AO	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**
	CPSO	2.54E-05	0.0017061	0.0031528	9.97E-13	0.49623	1.8854	2.27E-07	0.0031208	0.011286
	SSA	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**

续表 4　低维、高维函数上的性能比较

函数	算法	dim=30			dim=100			dim=500		
		最优解	平均值	标准差	最优解	平均值	标准差	最优解	平均值	标准差
f_8	MPA	3.53E-13	1.42E-12	1.16E-12	7.10E-12	4.60E-11	2.34E-11	2.22E-10	4.30E-10	1.29E-10
	EGolden-SWOA	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**	**8.88E-16**	**8.88E-16**	**0**
f_9	DEtanAO	**2.81E-32**	6.72E-08	2.26E-07	**1.69E-32**	3.17E-09	1.20E-08	**1.39E-32**	1.33E-10	5.87E-10
	tanAO	2.66E-12	7.58E-07	1.51E-06	2.53E-09	2.95E-07	4.86E-07	3.59E-10	1.89E-07	4.01E-07
	AO	6.56E-09	4.38E-06	9.48E-06	3.82E-09	3.07E-06	7.42E-06	1.09E-08	6.15E-07	7.44E-07
	CPSO	1.11E-05	0.68033	2.1691	8.34E-06	0.6621	2.1033	0.0001318	0.41022	1.6491
	SSA	8.60E-18	**3.65E-12**	**1.81E-11**	1.37E-10	7.40E-08	3.43E-07	2.49E-12	5.62E-08	1.10E-07
	MPA	1.22E-09	5.69E-09	3.34E-09	0.0236108	0.0441271	0.0122656	0.3516941	0.4089202	0.0289135
	EGolden-SWOA	6.15E-14	4.88E-08	2.43E-07	1.24E-14	**8.57E-10**	**3.28E-09**	8.56E-14	**3.11E-11**	**6.39E-11**

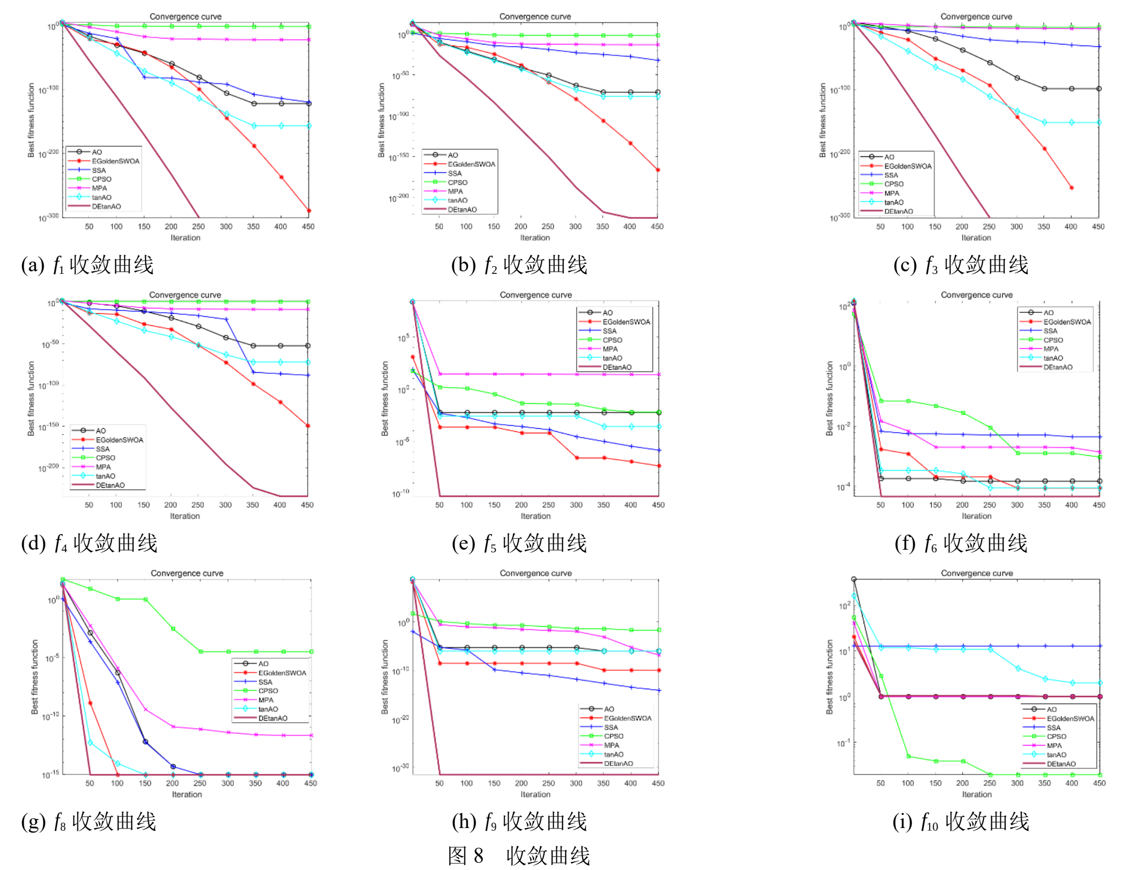
表 5　固定维数函数上的性能比较

Tab. 5　Performance comparison on fixed dimensional functions

算法	f_{10}			f_{11}			f_{12}		
	最优解	平均值	标准差	最优解	平均值	标准差	最优解	平均值	标准差
DEtanAO	**0.9980038**	2.6384379	2.895985	**0.3978874**	**0.3978881**	3.96E-06	**3**	3.0038	0.018132
tanAO	**0.9980038**	3.7318068	4.590161	**0.3979267**	0.4022246	0.0131830	3.0003929	3.3083947	0.414067
AO	**0.9980038**	2.7607481	3.459285	0.3978936	**0.3980834**	0.0001897	3.00003146	3.03204623	0.033251
CPSO	3.29E-05	0.13484	0.73388	2.25E-08	0.25729	1.4014	3.29E-06	0.7202	1.8772
SSA	**0.99800384**	3.0121167	4.260976	0.3978874	0.39788736	0	**3**	4	4.929503
MPA	**0.99800384**	**0.9980038**	**1.51E-16**	**0.3978874**	**0.39788736**	**1.23E-14**	**3**	**3**	**1.64E-15**
EGolden-SWOA	**0.99800384**	**0.9980038**	3.50E-12	0.3978875	0.3979736	0.0001675	3.00000002	3.00000887	1.11E-05

3.5　收敛曲线分析

算法的优劣可以通过其收敛曲线直观地表现出来，收敛曲线展示了算法的收敛速度和陷入局部最优值的次数。图 8(a)~(i)列出了 DEtanAO、AO、tanAO、CPSO、SSA、MPA、EGolden-SWOA 共 7 种算法在 30 维情况下，对上述 12 个基准函数的收敛曲线对比图。观察上述图中各算法的收敛曲线可以看出 DEtanAO 的收敛速度在整个迭代过程中都快于其他 6 种算法，收敛精度也是这 7 种算法中最好的，这不仅说明 DEtanAO 对比其他算法有更好的全局探索能力而且不易陷入局部最优解，平衡了全局探索能力和局部开发能力。其中为验证改进策略的有效性，将仅用切线飞行策略替换莱维飞行策略的算法记为 tanAO，从收敛曲线图中可知，tanAO 比基本的 AO 算法在收敛速度上有明显提升，观察 f_8 收敛曲线图可知 tanAO 在仅 150 次迭代就达到了最优值，而基本 AO 算法在 250 次迭代才达到最优值，tanAO 的寻优精度和速度比 AO 有较大幅度提升。



(a) f_1 收敛曲线　(b) f_2 收敛曲线　(c) f_3 收敛曲线

(d) f_4 收敛曲线　(e) f_5 收敛曲线　(f) f_6 收敛曲线

(g) f_8 收敛曲线　(h) f_9 收敛曲线　(i) f_{10} 收敛曲线

图 8　收敛曲线

Fig. 8　Convergence curve

综上所述，本文提出的 DEtanAO 算法不管在低维、高维还是固定维度的函数下都具有较好的寻优性能，其求解精度、收敛速度和鲁棒性均优于与之对比的其他六种算法，有着明显的求解优势。

3.6 Wilcoxon 秩和检验

Wilcoxon[16]秩和检验是一种非参数统计检验方法，用于判断比较 DEtanAO 算法与其他算法是否有显著性区别，其结果分析如表 6 所示，其中“p”表示检验结果，“h”表示显著性判断结果。当 $p<0.05$ 时，h 显示为“1”，表示 DEtanAO 算法的显著性强于其他算法；当 $p>0.05$ 时，h 显示为“0”，表示 DEtanAO 算法的显著性弱于其他算法；当 P 显示为“N/A”时，表示无法进行显著性检验，DEtanAO 算法显著性可能与其他算法相同。

将 DEtanAO 与 AO、tanAO、CPSO、SSA、MPA、EGolden-SWOA 独立运行 30 次的平均值进行 Wilcoxon 秩和检验，从表 6 中的统计结果可以看出，大部分的 p 值均小于 0.05，表明 DEtanAO 与其他算法存在显著差异，并且 DEtanAO 的性能在多个函数上优于对比算法。

表 6 基准函数 Wilcoxon 秩和检验结果

Tab. 6 Wilcoxon rank sum test results of reference functions

函数	AO		tanAO		EGoldenSWOA		MPA		SSA		CPSO	
	p	h	p	h	p	h	p	h	p	h	p	h
f_1	1.21E-12	1	1.21E-12	1	N/A	0	1.21E-12	1	1.21E-12	1	1.21178E-12	1
f_2	3.02E-11	1	3.02E-11	1	1.61323E-10	1	3.02E-11	1	3.02E-11	1	3.01986E-11	1
f_3	1.21E-12	1	1.21E-12	1	N/A	0	1.21E-12	1	1.21E-12	1	1.21178E-12	1
f_4	3.02E-11	1	3.02E-11	1	3.01986E-11	1	3.02E-11	1	3.02E-11	1	3.01986E-11	1
f_5	1.41E-09	1	1.29E-09	1	0.077271976	0	3.02E-11	1	2.13E-05	1	2.43863E-09	1
f_6	0.099258	0	0.145319	0	0.016284809	1	3.69E-11	1	8.48E-09	1	5.53286E-08	1
f_7	0.333711	0	0.333711	0	0.333710696	0	0.333711	0	0.333711	0	1.72025E-12	1
f_8	N/A	0	N/A	0	N/A	0	1.21E-12	1	N/A	0	1.21178E-12	1
f_9	3.02E-11	1	3.02E-11	1	2.37591E-07	1	8.15E-11	1	9.51E-06	1	3.01797E-11	1
f_{10}	0.501102	0	0.841786	0	7.18381E-05	1	5.78E-08	1	0.042542	1	1.72466E-07	1
f_{11}	3.16E-12	1	3.16E-12	1	3.15782E-12	1	0.024657	1	0.081493	0	3.22767E-07	1
f_{12}	1.36E-08	1	1.02E-10	1	1.01948E-07	1	0.186128	0	0.509236	0	1.01948E-07	1
1/0	8/4		8/4		7/5		10/2		8/4		12/0	

3.7 工程优化问题应用

为验证 DEtanAO 算法的前沿性以及在实际工程中的优越性，选取了压力容器设计优化问题，并通过与上述对比算法的改进算法如融合正余弦和柯西变异的麻雀搜索算法(SCSSA)[17]、混合策略改进的麻雀搜索算法(MSSSA)[18]基于分段式随机惯性权重和最优反馈机制的鲸鱼优化算法(FWOA)[19]、高斯量子粒子群算法(G-QPSO)[20]进行验证比较。压力容器设计问题[21]是一个经典工程优化设计问题，目的是通过优化圆柱容器身长 L、圆柱容器内径 R、圆柱体壁厚 T_s 和头部壁厚 T_h 变量，使压力容器制作成本达到最小。问题的目标函数、约束条件和自变量取值范围如下所示。

$$X=[x_1,x_2,x_3,x_4]=[T_a,T_b,R,L]$$

$$Min f(x)=0.6224x_1x_2x_3+1.7781x_2x_3^2+3.1661x_4x_1^2+19.84x_3x_1^2$$

$$s.t\begin{cases} g_1(x)=-x_1+0.0193x_3\le 0 \\ g_2(x)=-x_2+0.00954x_3\le 0 \\ g_3(x)=-\pi x_3^2-\dfrac{4}{3}\pi x_3^2+1296000\le 0 \\ g_4(x)=-x_4-240\le 0 \\ 0\le x_1\le 100, i=1,2 \\ 0\le x_1\le 200, i=3,4 \end{cases} \tag{17}$$

求解结果见表 7，将四种改进算法的求解结果与 DEtanAO 算法的求解结果进行比较。从表中可以看出在求解压力容器设计问题时 DEtanAO 算法的最优解和平均值均小于其他对比算法，说明 DEtanAO 算法在此类优化问题上的求解精度较高，稳定性也能达到较好的数值，整体验证了 DEtanAO 算法的前沿性和优越性。

4 结束语

本文提出一种融合差分进化和切线飞行的天鹰优化器，通过差分进化策略，改善了原算法过度依赖天鹰根据不同攻击方式的数学公式进行位置更新的状况，利用切线飞行策略，有利于算法在随机游走的过程中出现更大概率的大跨步，使得算法跳出局部最优，提高了算法的寻优质量。通过 12 个标准基准函数、高维函数、Wilcoxon 秩和检验以及在工程优化问题上的应用的结果显示，DEtanAO 具有较强的跳出局部最优的能力、更快的收敛速度以及更高的收敛精度。下一步考虑将 DEtanAO 算法应用到其他实际问题的建模中，比如机器学习各种预测分类算法中超参数的优化，并进一步将优化后的机器学习算法用来预测量化投资中股价变动的趋势，以扩展本文所提算法在金融等领域的应用范围。

表 7 压力容器设计问题求解结果

Tab. 7 Pressure vessel design problem solving results

算法	最小值	平均值	标准差
DEtanAO	**5862.69**	**6081.35**	**2.15E+02**
FWOA	5886.97	-	-
SCSSA	8050.92	-	-
MSSSA	5897.15	6615.81	5.53E+02
G-QPSO	6059.72	6342.45	4.48E+02

参考文献：

[1] Yu Z, Li Q, Feng Y, *et al.* Hierarchical economic load dispatch based on chaotic-particle swarm optimization [C]// Ninth International Conference on Natural Computation. IEEE, 2013: 517-521.

[2] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm [J]. Systems Science & Control Engineering An Open Access Journal, 2020, 8 (1): 22-34.

[3] Faramarzi A, Heidarinejad M, Mirjalili S, *et al.* Marine predators algorithm: A nature-inspired metaheuristic [J]. Expert Systems with Applications, 2020, 152: 113377.

[4] 肖子雅, 刘升. 精英反向黄金正弦鲸鱼算法及其工程优化研究 [J]. 电子学报, 2019, 47 (10): 2177-2186. (Xiao Ziyao, Liu Sheng. Study on elite opposition-based golden-sine whale optimization algorithm and its application of project optimization [J]. Acta Electronica Sinica, 2019, 47

(10): 2177-2186.)

[5] Abualigah L, Yousri D, Elaziz M A, *et al.* Matlab Code of Aquila Optimizer: A novel meta-heuristic optimization algorithm [J]. Computers & Industrial Engineering, 2021, 157: 107250.

[6] Wang S, Jia H, Abualigah L, *et al.* An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems [J]. Processes, 2021, 9 (9): 1551.

[7] Mahajan, S. , Abualigah, L. , Pandit, A. K. *et al.* Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks [J]. Soft Computing, 2022, 26, 4863–4881.

[8] Shicheng Wang, Jiawei Ma, Wei Li, *et al.* An optimal configuration for hybrid SOFC, gas turbine, and Proton Exchange Membrane Electrolyzer using a developed Aquila Optimizer [J]. International Journal of Hydrogen Energy, 2022, 47 (14) 8943-8955.

[9] Eric Ofori-Ntow Jnr, Yao Yevenyo Ziggah, Maria Joao Rodrigues, *et al.* A hybrid chaotic-based discrete wavelet transform and Aquila optimisation tuned-artificial neural network approach for wind speed prediction [J]. Results in Engineering, 2022, 14, 2590-1230.

[10] Layeb, A. Tangent search algorithm for solving optimization problems [J/OL]. Neural Comput & Applic, 2022, 2022 (34) , 8853–8884. https://doi. org/10. 1007/s00521-022-06908-z.

[11] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.

[12] Shao X, Chen Z, Lin X. Resolution of multicomponent overlapping chromatogram using an immune algorithm and genetic algorithm [J]. Chemometrics & Intelligent Laboratory Systems, 2000, 50 (1): 91-99.

[13] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.

[14] Reynolds A M, Smith A D, Reynolds D R, *et al.* Honeybees perform optimal scale-free searching flights when attempting to locate a food source [J]. Journal of Experimental Biology, 2007, 210 (21): 3763-3770.

[15] Mantegna Rosario Nunzio. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes [J]. Physical Review, 1994, 49 (5): 4677-4683.

[16] Wilcoxon F. Individual comparison by ranking methods [J]. Biometrics, 1945, 1: 80-83.

[17] 李爱莲，全凌翔，崔桂梅，解韶峰. 融合正余弦和柯西变异的麻雀搜索算法 [J]. 计算机工程与应用, 2022, 58 (03): 91-99. (Li Ailian, Quan Linxiang, Cui Guimei, Xie Shaofeng. Sparrow Search Algorithm Combining Sine-Cosine and Cauchy Mutation [J]. Computer Engineering and Applications, 2022, 58 (03): 91-99.)

[18] 张伟康，刘升，任春慧. 混合策略改进的麻雀搜索算法 [J]. 计算机工程与应用, 2021, 57 (24): 74-82. (Zhang Weikang, Liu Sheng, Ren Chunhui. Mixed Strategy Improved Sparrow Search Algorithm [J]. Computer Engineering and Applications, 2021, 57 (24): 74-82.)

[19] 刘景森，马义想，李煜. 改进鲸鱼算法求解工程设计优化问题 [J]. 计算机集成制造系统, 2021, 27 (07): 1884-1897. (Liu Jinsen, Ma Yixiang, Li Yu. Improved whale algorithm for solving engineering design optimization problems [J]. Computer integrated manufacturing system, 2021, 27 (07): 1884-1897.)

[20] Leandro dos Santos Coelho. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems [J]. Expert Systems With Applications, 2009, 37 (2): 1676-1683.

[21] Seyedali Mirjalili and Seyed Mohammad Mirjalili and Abdolreza Hatamlou. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization [J]. Neural Computing and Applications, 2016, 27 (2): 495-513.

[22] 马威强，高永琪，赵苗. 基于全局最优和差分变异的头脑风暴优化算法 [J]. 系统工程与电子技术, 2022, 44 (1): 270-278. (Ma Weiqiang, Gao Yongqi, Zhao Miao. Global-best difference-mutation brain storm optimization algorithm [J]. Systems Engineering and Electronics, 2022, 44 (1): 270-278.)